IA-32 Intel® Architecture Software Developer's Manual

Documentation Changes

December 2002

Notice: The IA-32 Intel® Architecture may contain design defects or errors known as errata that may cause the product to deviate from published specifications. Current characterized errata are documented in this specification update.

Document Number: 252046

Contents

Revision History

Version	Description	Date
-001	Initial Release	November 2002
-002	Added 1-10 Documentation Changes. Removed old Documentation Changes items that already have been incorporated in the published Software Developer's manual	December 2002

Preface

This document is an update to the specifications contained in the Affected Documents/Related Documents table below. This document is a compilation of documentation changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

Information types defined in Nomenclature are consolidated into the specification update and are no longer published in other documents.

This document may also contain information that was not previously published.

Affected Documents/Related Documents

Document Title	Document Number
IA-32 Intel® Architecture Software Developer's Manual: Volume 1, Basic Architecture	245470-008
IA-32 Intel® Architecture Software Developer's Manual: Volume 2, Instruction Set Reference	245471-008
IA-32 Intel® Architecture Software Developer's Manual: Volume 3, System Programming Guide	245472-008)

Nomenclature

Documentation Changes include typos, errors, or omissions from the current published specifications. These will be incorporated in any new release of the specification.

Summary Table of Changes

The following table indicates the errata, specification changes, specification clarifications, or documentation changes which apply to the IA-32 Intel Architecture. This table uses the following notations:

Codes Used in Summary Table

Change bar to left of table row indicates this erratum is either new or modified from the previous version of the document.

Documentation Changes

Number	DOCUMENTATION CHANGES
1	Description of FCOMI/FCOMIP Updated
2	Note Added to POP CS Instruction Entry in Table B-10
3	Information Added on resource_stall and WC_Buffer
4	VIF Information Added to CLI Discussion
5	Switching the L1 Data Cache between Adaptive and Shared Mode
6	References to bits Gen and Len Deleted
7	FTW Storage Location Corrected
8	More Information on Invalid TSS Conditions Provided
9	Memory Storage Location of x87 FPU Data Registers Specified for FSAVE
10	Valid Interrupt Vector Discussion Has More Explanation

Documentation Changes

All Documentation Changes will be incorporated into a future version of the IA-32 Intel® Architecture Software Developer's Manual.

1. Description of FCOMI/FCOMIP Updated

The *IA-32 Intel® Architecture Software Developer's Manual, Volume 2: Instruction Set Reference*, Chapter 3, Section: *FCOMI/FCOMIP/ FUCOMI/FUCOMIP-Compare Floating Point Values and Set EFLAGS* currently states:

The FUCOMI/FUCOMIP instructions always clear the OF flag in the EFLAGS register (regardless of whether an invalid-operation exception is detected); the FCOMI/FCOMIP instructions do not clear the OF flag.

It should state:

The FCOMI/FCOMIP and FUCOMI/FUCOMIP instructions clear the OF flag in the EFLAGS register (regardless of whether an invalid-operation exception is detected.

2. Note Added to POP CS Instruction Entry in Table B-10

The *IA-32 Intel® Architecture Software Developer's Manual, Volume 2: Instruction Set Reference*, Appendix B: *Instructions Formats and Encodings*, Table B-10: *General Purpose Instruction Formats and Encodings* has been updated by adding a note to the POP instruction. The note addresses a common error condition. The note is shown in the table segment below.

Table B-10. General Purpose Instruction Formats and Encodings [table segment only, entire table not shown...]

POP – Pop a Segment Register from the Stack (Note: CS cannot be sreg2 in this usage.)	
segment register DS, ES	000 sreg2 111
segment register SS	000 sreg2 111
segment register FS, GS	0000 1111: 10 sreg3 001

3. Information Added on resource_stall and WC_Buffer

The *IA-32 Intel® Architecture Software Developer's Manual, Volume 3: System Programming Guide*, Appendix A, Table 1 will now include the following information on resource_stall and WC_Buffer. The new data is shown in the table segments included below:

Table A-1. Pentium 4 and Intel Xeon Processor Performance Monitoring Events for Non-Retirement Counting [table segment only, entire table not shown...]

Event Name	Parameters	Parameter Value	Description
resource_stall			This event monitors the occurrence or latency of stalls in the Allocator.
	ESCR restrictions	MSR_ALF_ESCR0 MSR_ALF_ESCR1	
	Counter numbers per ESCR	ESCR0: 12, 13, 16 ESCR1: 14, 15, 17	
	ESCR Event Select	01H	ESCR[30:25]
	Event Masks	Bit 5: SBFULL	ESCR[24:9] A Stall due to lack of store buffers
	CCCR Select	01H	CCCR[15:13]
	Event Specific Notes		This event may not be supported in all models of the processor family.
WC_Buffer			This event counts Write Combining Buffer operations that are selected by the event mask.
	ESCR restrictions	MSR_DAC_ESCR0 MSR_DAC_ESCR1	
	Counter numbers per ESCR	ESCR0: 8, 9 ESCR1: 10, 11	
	ESCR Event Select	05H	ESCR[30:25]
	Event Masks	Bit 0: WCB_EVICTS 1: WCB_FULL _EVICT 2: WCB_HITM _EVICT	ESCR[24:9] WC Buffer evictions of all causes WC Buffer eviction: no WC buffer is available WC Buffer eviction: Store encountered a Hit Modified condition

4. VIF Information Added to CLI Discussion

The *IA-32 Intel® Architecture Software Developer's Manual, Volume 2: Instruction Set Reference,* Chapter 3, *Section: CLI-Clear Interrupt Flag* section now incorporates more complete information about the response of the VIF flag. The information is summarized in the following table (see the VIF line).

Table 3-5. Response to CLI

Flags	Real Mode	Protected Mode: Virtual Interrupts on, CLEAR VIF	Protected Mode: Virtual Interrupts off, CLEAR IF	Protected Mode: Virtual Interrupts off, GP Fault (PVI = 0)	Protected Mode: Virtual Interrupts off, GP Fault (CPL < 3)	vm86 mode: CLEAR IF	vm86 mode: GP Fault
PE	0	1	1	1	1	1	1
VM	X	0	1	0	0	1	1
CPL	X	3	<= IOPL	> IOPL	< 3	X	X
IOPL	X	< 3	X	X	< CPL	3	< 3
PVI	X	1	X	0	1	X	X
IF = 0	Y	N	Y	N	N	Y	N
VIF = 0	N	Y	N	N	N	N	N
#GP	N	N	N	Y	Y	N	Y

Notes:
X Don't care
N Action not taken
Y Action taken

5. Switching the L1 Data Cache between Adaptive and Shared Mode

The ability to switch the L1 Data Cache between adaptive and shared mode is now documented. In the *Intel Architecture Software Developer's Manual*, *Volume 2: Instruction Set Reference*, Chapter 3, Table 3-7; this capability impacts the description of ECX. See the EAX=1, ECX[10] in the following table segment:

Table 3-7. Information Returned By CPUID Instruction [table segment only, entire table is not shown...]

Initial EAX Value		Information Provided About the Processor
		Basic CPUID Information
1H	EAX	Version Information (Type, Family, Model, and Stepping ID)
	EBX	Bits 7-0: Brand Index Bits 15-8: CLFLUSH line size. (Value ∗ 8 = cache line size in bytes) Bits 23-16: Number of logical processors per physical processor. Bits 31-24: Local APIC ID
	ECX	Bits 9-0: Reserved **Bit 10: If this bit is 1, the L1 Data Cache may be placed in adaptive mode or shared mode. Mode selection is determined by Bit 24 of IA32_MISC_ENABLE (see Volume 3, Appendix B, Table B-1). If ECX[10] is 0, the ability to change the L1 Data Cache mode is not supported.** Bits 63-11: Reserved
	EDX	Feature Information (see Figure 3-4 and Table 3-10)

Supporting information has also been added to the *IA-32 Intel® Architecture Software Developer's Manual, Volume 3: System Programming Guide*, Appendix B, Table B1. The description IA32_MISC_ENABLE[24] has been changed. This information has been provided in the table segment shown:

Table B-1. Special Fields within Instruction Encodings [table segment only...]

Register Address		Register Name Fields and Flags	Shared/ Unique[1]	Bit Description
Hex	Dec			
1A0H	416	IA32_MISC_ENABLE	Shared	Enable Miscellaneous Processor Features. (R/W) Allows a variety of processor functions to be enabled and disabled.
		0		**Fast-Strings Enable.** When set, the fast-strings feature on the Pentium 4 processor is enabled(default); when clear, fast-strings are disabled.
		1		**Reserved.**
		2		**x87 FPU Fopcode Compatibility Mode Enable.** When set, fopcode compatibility mode is enabled; when clear (default), mode is disabled. See “Fopcode Compatibility Mode“ in Chapter 8 of the *IA-32 Intel Architecture Software Developer’s Manual, Volume 1*.
		3		**Thermal Monitor Enable.** When set, clock modulation controlled by the processor’s internal thermal sensor is enabled; when clear (default), automatic clock modulation is disabled. (See Section 13.14.2., “Automatic Thermal Monitor”.)

Table B-1. Special Fields within Instruction Encodings [table segment only...]				
Register Address		Register Name Fields and Flags	Shared/ Unique[1]	Bit Description
Hex	Dec			
		4		**Split-Lock Disable.** This debug feature is specific to the Pentium 4. When set, the bit causes an #AC exception to be issued instead of a split-lock cycle. Operating systems that set this bit must align system structures to avoid split-lock scenarios. When the bit is clear (the default), normal split-locks are issued to the bus.
		5		**Reserved.**
		6		**Third-Level Cache Disable.** (R/W) When set, the third-level cache is disabled; when clear (default) the third-level cache is enabled. This flag is reserved for processors that do not have a third-level cache. Note that this bit controls only the third-level cache, and then only if overall caching is enabled through the CD flag of control register CR0, the page-level cache controls, and/or the MTRRs (see Section 10.5.4., “Disabling and Enabling the L3 Cache”).
		7		**Performance Monitoring Available.** (R) When set, performance monitoring is enabled; when clear, performance monitoring is disabled.
		8		**Suppress Lock Enable.** When set assert on of lock on the bus is suppressed during a Split Lock access. when clear (default) does not suppress lock.
		9		**Prefetch Queue Disable.** When set disables the prefetch queue. When clear (default) the prefetch queue is enabled.
		10		**FERR# Interrupt Reporting Enable. (R/W)** When set, interrupt reporting through the FERR# pin is enabled; when clear, this interrupt reporting function is disabled. When this flag is set and the processor is in the stop-clock state (STPCLK# is asserted), asserting the FERR# pin signals to the processor that an interrupt (such as, INIT#, BINIT#, INTR, NMI, SMI#, or RESET#) is pending and that the processor should return to normal operation to handle the interrupt. This flag does not affect the normal operation of the FERR# pin (to indicate an unmasked floating-point error) when the STPCLK# pin is not asserted.
		11		**Branch Trace Storage Unavailable (BTS_UNAVILABLE).** (R) When set, the processor does not support branch trace storage (BTS); when clear, BTS is supported.

Table B-1. Special Fields within Instruction Encodings [table segment only...]				
Register Address		Register Name Fields and Flags	Shared/ Unique[1]	Bit Description
Hex	Dec			
		12		**Precise Event Based Sampling Unavailable (PEBS_UNAVILABLE).** (R) When set, the processor does not support precise event-based sampling (PEBS); when clear, PEBS is supported.
		23:13		**Reserved.**
		24		**L1 Data Cache Context Mode (R/W).** When set to 1, this bit places the L1 Data Cache into shared mode. When set to 0 (the default), this bit places the L1 Data Cache into adaptive mode. In adaptive mode, the Page Directory Base Register contained in CR3 must be identical across all logical processors. Note: If the Context ID feature flag, ECX[10], is not set to 1 after executing CPUID with EAX = 1; the ability to switch modes is not supported and the BIOS must not alter the contents of IA32_MISC_ENABLE[24].
		63:25		**Reserved.**

6. References to Bits Gen and Len Deleted

The *IA-32 Intel® Architecture Software Developer's Manual, Volume 3: System Programming Guide*, Chapter 15, Table 15-2,*Debug Exception Condition* referenced bits Gen and Len. They don't exist in this context; this error has been removed. The corrected table is shown:

Table 15-2. Debug Exception Conditions			
Debug or Breakpoint Condition	DR6 Flags Tested	DR7 Flags Tested	Exception Class
Single-step trap	BS = 1		Trap
Instruction breakpoint, at addresses defined by DR*n* and LEN*n*	B*n* = 1 and (G*n* or L*n* = 1)	R/Wn = 0	Fault
Data write breakpoint, at addresses defined by DR*n* and LEN*n*	B*n* = 1 and (G*n* or L*n* = 1)	R/Wn = 1	Trap
I/O read or write breakpoint, at addresses defined by DR*n* and LEN*n*	B*n* = 1 and (G*n* or L*n* = 1)	R/Wn = 2	Trap
Data read or write (but not instruction fetches), at addresses defined by DR*n* and LEN*n*	B*n* = 1 and (G*n* or L*n* = 1)	R/Wn = 3	Trap
General detect fault, resulting from an attempt to modify debug registers (usually in conjunction with in-circuit emulation)	BD = 1		Fault
Task switch	BT = 1		Trap

7. FTW Storage Location Corrected

The *IA-32 Intel® Architecture Software Developer's Manual, Volume 2: Instruction Set Reference*, Chapter 3, Section: *FXSAVE - Save x87 FPU, MMX, SSE, and SSE2 State* incorrectly indicates the memory storage location of the x87 FPU Tag Word (FTW). The storage location has been corrected to indicate a byte 4 offset (instead of a byte 5 offset).

A partial representation of the corrected table is shown below.

Table 3-14. Layout of FXSAVE and FXRSTOR Memory Region [table segment only, entire table is not shown...]

15 14	13 12	11 10	9 8	7 6	5	4	3 2	1 0	
Rsrvd	CS	FPU IP		FOP		**FTW**	FSW	FCW	**0**
MXCSR_MASK		MXCSR		Rsrvd	DS		FPU DP		**16**
Reserved		ST0/MM0							**32**
Reserved		ST1/MM1							**48**
Reserved		ST2/MM2							**64**
Reserved		ST3/MM3							**80**
Reserved		ST4/MM4							**96**
Reserved		ST5/MM5							**112**
Reserved		ST6/MM6							**128**
Reserved		ST7/MM7							**144**
XMM0									**160**

8. More Information on Invalid TSS Conditions Provided

In the *A-32 Intel® Architecture Software Developer's Manual, Volume 3: System Programming Guide*, Chapter 5, Section: *Interrupt 10 - Invalid TSS Exception (#TS)*, Table 5-6 has been modified to include a more complete list of invalid conditions. The impacted part of the table is reproduced below with additions indicated by change bars.

Table 5-6. Invalid TSS Conditions [table segment only, entire table not shown...]

Error Code Index	Invalid Condition
TSS segment selector index	TSS segment limit less than 67H for 32-bit TSS or less than 2CH for 16-bit TSS
TSS segment selector index	During an IRET task switch, the TI flag in the TSS segment selector indicates the LDT
TSS segment selector index	During an IRET task switch, the TSS segment selector exceeds descriptor table limit
TSS segment selector index	During an IRET task switch, the busy flag in the TSS descriptor indicates an inactive task
LDT segment selector index	Invalid LDT or LDT not present

In the *IA-32 Intel® Architecture Software Developer's Manual, Volume 3: System Programming Guide*, Chapter 5, Section: *Interrupt 13 – General Protection Exception (#GP)* includes a large bulleted list. The list has been corrected. The corrected portion of the list is shown below (see the change bars).

- Loading the CS register with a segment selector for a data segment or a null segment selector.
- Accessing memory using the DS, ES, FS, or GS register when it contains a null segment selector.
- Switching to a busy task during a call or jump to a TSS.
- Using a segment selector on a non-IRET task switch that points to a TSS descriptor in the current LDT. TSS descriptors can only reside in the GDT. This condition causes a #TS exception during an IRET task switch.
- Violating any of the privilege rules described in Chapter 4, *Protection*.
- Exceeding the instruction length limit of 15 bytes (this only can occur when redundant prefixes are placed before an instruction).

9. Specified for FSAVE

In the *IA-32 Intel® Architecture Software Developer's Manual, Volume 1: Basic Architecture*, Chapter 8, Figures 8-9 and 8-10 have been updated to include a short note about storage of the X87 data registers during the operation of FSAVE. Both updated figures are shown below:

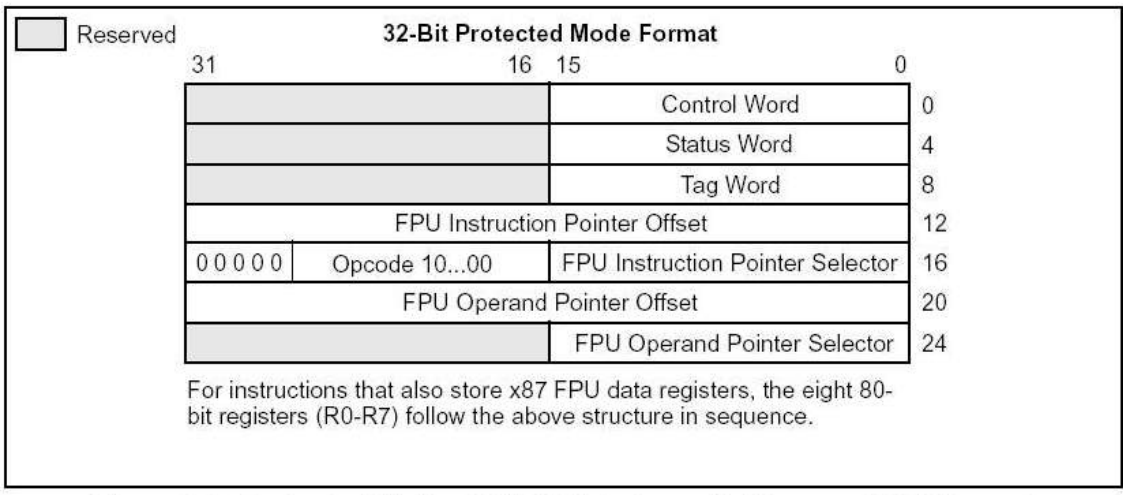

Figure 8-9. Protected Mode x87 FPU State Image in Memory, 32-Bit Format

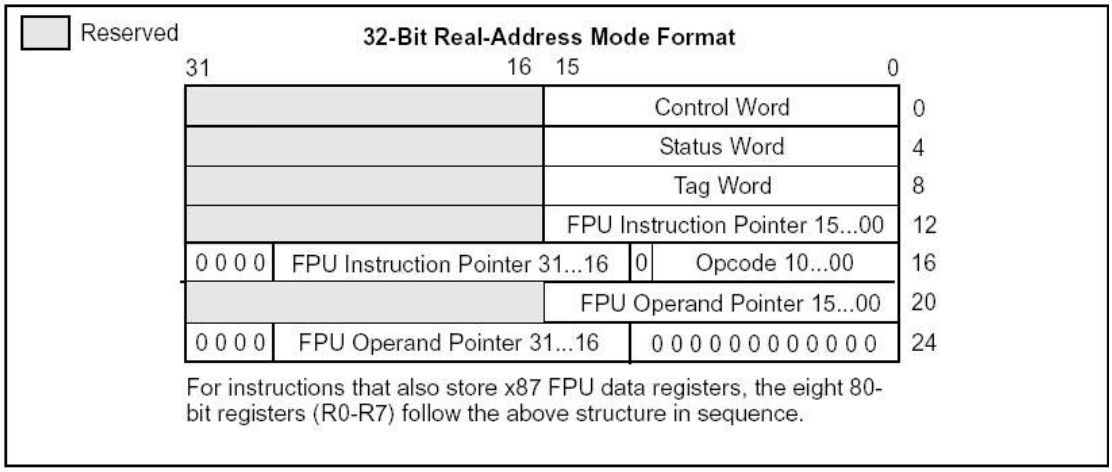

Figure 8-10. Real Mode x87 FPU State Image in Memory, 32-Bit Format

10. Valid Interrupt Vector Discussion Has More Explanation

In the *IA-32 Intel® Architecture Software Developer's Manual, Volume. 3: System Programming Guide*, Chapter 8, Section 8.5.2 has been updated to include more information. See the data below. Change bars note the updated data.

Section 8.5.2 Valid Interrupt Vectors

The IA-32 architecture defines 256 vector numbers, ranging from 0 through 255 (see Section 5.2., *Exception and Interrupt Vectors)*. The local and I/O APICs support 240 of these vectors (in the range of 16 to 255) as valid interrupts.

When an interrupt vector in the range of 0 to 15 is sent or received through the local APIC, the APIC indicates an illegal vector in its Error Status Register [see Section 8.5.3., *Error Handling*]. The IA-32 architecture reserves vectors 16 through 31 for predefined interrupts, exceptions, and Intel-reserved encodings (see Table 5-1); however, the local APIC does not treat vectors in this range as illegal.

When an illegal vector value (0 to 15) is written to an LVT entry and the delivery mode is Fixed (bits 8-11 equal 0), the APIC may signal an illegal vector error, without regard to whether the mask bit is set or whether an interrupt is actually seen on the input